A4 cont'd.

modifying each reply request with a portion of the collected data associated with the portion of a prior HTTP request.

[0047] **(Amended)** In general, in one aspect, the invention comprises a tool for testing and debugging a web application. The tool comprises means for sending a requested execution mode to the debugging controller, means for determining an execution mode of a server, means for comparing the requested execution mode and the execution mode of the server, means for switching the execution mode of the server to match the requested execution mode, means for forwarding the HTTP request to the execution server, and means for running and modifying each reply request with a portion of the collected data associated with the portion of a prior HTTP request.

A5

[0056] **(Amended)** The present invention allows the user to transparently replay a HTTP request while simultaneously switching the server to debugging mode and opening the debugging client. Once this occurs the error can be readily identified through source level debugging using the debugging client. Once the error has been identified and fixed, the present invention allows the user to transparently switch back to normal execution mode. In normal execution mode, the present invention allows the user to replay the HTTP request to verify that the error was fixed.

A6

[0063] **(Amended)** The client-side component **18** also includes a debugging controller **19** that controls the execution mode of the server *i.e.,* normal or debugging, and the mechanism for allowing transparent switching between the two modes. The normal mode corresponds to regular execution of the execution server **10**. The debugging mode corresponds to executing the HTTP requests with a debugging session on the execution server **10**. When the client-side component issues a replay request, it first contacts the debugging controller **19** to check whether the execution server **10** which the request is intended to run on is in the desired mode. If not, the client-side component **18** issues a request to the debugging controller **19** to switching the mode of the execution server **10**. Once the client-side component **18** receives confirmation from the debugging controller **19** that the server is running in the desired mode, it issues the request.

A7

[0073] **(Amended)** The "Replay" action invokes the debugging controller **19** to detect if the execution server **10** upon which the HTTP request was originally executed on is running. If the execution server **10** is not running, the execution server **10** is started. The execution server **10** is started in the normal mode if button **35** is toggled. The execution server **10** is started in the debugging mode if the button **33** is toggled. If the execution server **10** is running in normal mode and button **33** is toggled, the execution server **10** is switching to debugging mode. If the execution server **10** is running in debugging mode and button **35** is toggled, the execution server **10** is switched to normal mode. The HTTP request is then sent to the execution server **10**.

A8

[0074] **(Amended)** The "Edit and Replay" action brings up a dialog, which allows the user to modify the HTTP request data prior to causing it to be re-processed by the execution server **10**. Additionally, the user can modify which execution server **10** the user wants to execute on through the dialog. When the user completes the editing of the HTTP request and sends the HTTP request to the execution server **10**, the debugging controller **19** intercepts and examines the HTTP request. If the execution server **10** upon which the HTTP request is to be run is not running, the execution server **10** is started. The execution server **10** is started in the normal mode if button **35** is toggled. The execution server **10** is started in the debugging mode if the button **33** is toggled. If the execution server **10** is running in normal mode and button **33** is toggled, the execution server **10** is switched to debugging mode. If the execution server **10** is running in debugging mode and button **35** is toggled, the execution server **10** is switched to normal mode. The modified HTTP request is then sent to the execution server **10**.

A9

[0076] **(Amended)** "Replay sequence" causes the HTTP requests described by each sequence folder to be replayed one at a time in sequence. With "Replay sequence (step)," the output from each request is shown before moving on to the next request. With "Replay sequence (to end)," only the output of the last request is shown. To execute the sequence, a thread is started which attempts to execute each request in order, waiting for one to succeed before executing the next one. Each individual

A10 cont

request has have a timeout, which can be some default number or can be set by the user. If any individual request fails, the output of that request is displayed to the user, and the user is notified of the problem. Otherwise, the output of the last request in the sequence is displayed. For each replay request within the sequence the debugging controller **19** intercepts and examines the HTTP request. If the execution server **10**, upon which the HTTP request is to be run is not running, the execution server **10** is started. The execution server **10** is started in the normal mode if button **35** is toggled. The execution server **10** is started in the debugging mode if the button **33** is toggled. If the execution server **10** is running in normal mode and button **33** is toggled, the execution server **10** is switched to debugging mode. If the execution server **10** is running in debugging mode and button **35** is toggled, the execution server **10** is switched to normal mode. The HTTP request is then sent to the execution server **10**.

A10 cncld.

[0081] **(Amended)** The debugging controller **19**, based on the aforementioned information, may send a command **57** to execution server **10**, to start the execution server **10** in the mode specified in the GUI **22**. Based on the command **57**, the execution server **6** may be exited and may be restarted. In one implementation, debugging, for portions of the execution server **10**, may be activated without shutting down the entire execution server **10**. Once the mode is selected, the debugging controller **19** sends a confirmation to the client-side component **16**. The client-side **16** subsequently sends the request to the execution server **10** via an internal HTTP server **12**.

A11

[0084] **(Amended)** In one implementation, the debugging controller **19** uses a "Server Integration" API for controlling the server execution functionality. In this implementation, each execution server provides an implementation of the API called a server integration plugin. The server integration plugin uses proprietary methods for starting, stopping, and turning debugging on/off on the execution server. When the debugging controller receives a request to turn on debugging, the server integration plugin turns on debugging on the execution server using proprietary methods.

A12